GCT535 Sound Technology for Multimedia

JUCE and Homework 3 Tutorial

Minsuk Choi
Jaekwon Im

Introduction

JUCE is a **cross-platform** C++ application framework.

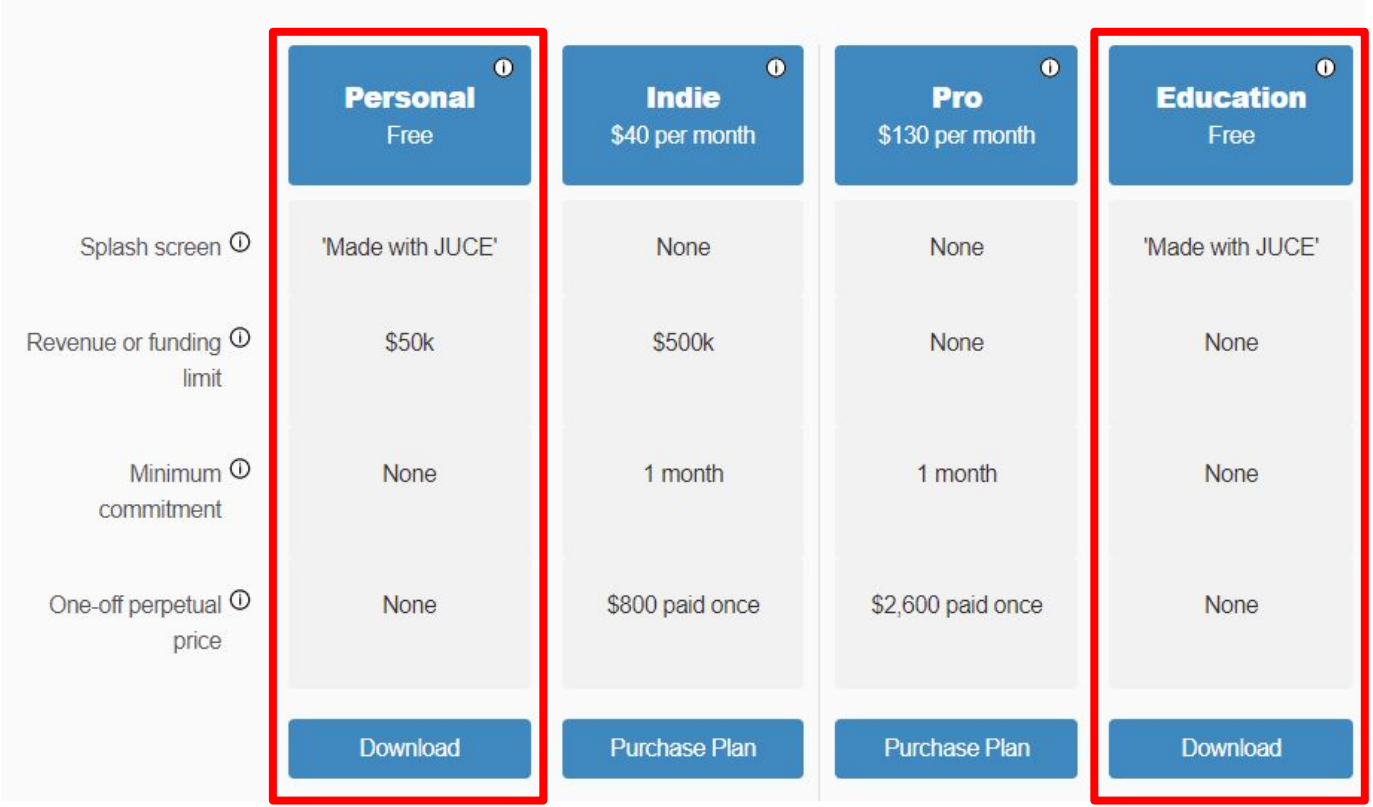
- Compile and run on identically on Windows, macOS, Linux, iOS and Android.

Specialized in **audio programming** and GUI.

- Capable of working with various audio files (WAV, AIFF, MP3, …), audio devices, MIDI, DSP algorithms and more.

Free for personal and educational usage.

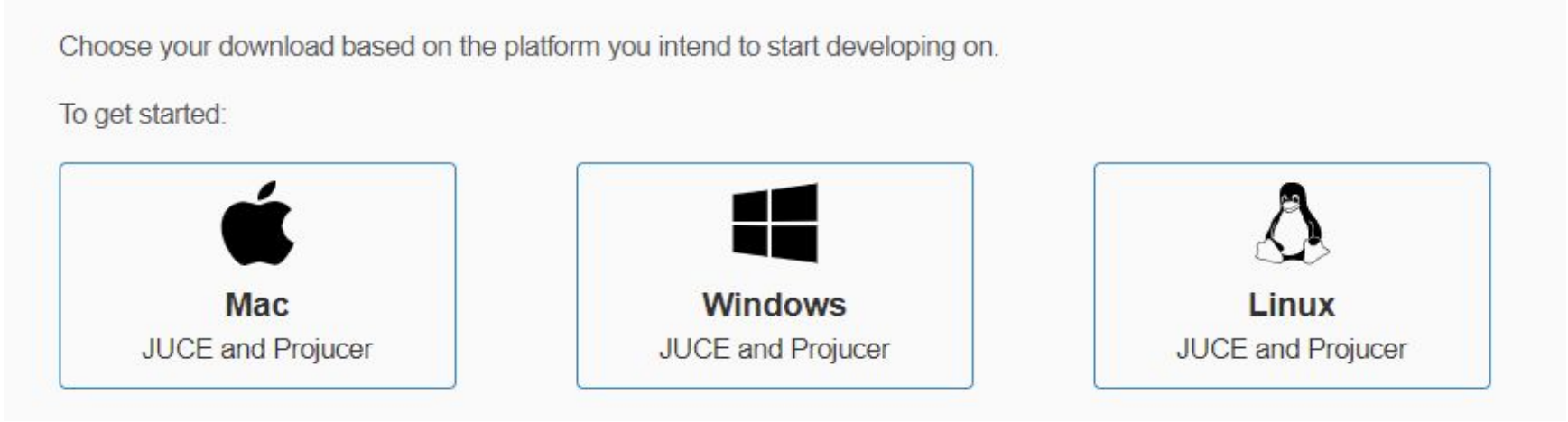
Installation



	Personal Free	Indie $40 per month	Pro $130 per month	Education Free
Splash screen	'Made with JUCE'	None	None	'Made with JUCE'
Revenue or funding limit	$50k	$500k	None	None
Minimum commitment	None	1 month	1 month	None
One-off perpetual price	None	$800 paid once	$2,600 paid once	None
	Download	Purchase Plan	Purchase Plan	Download

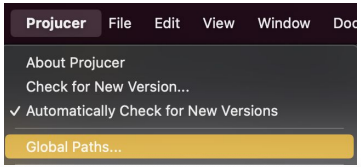
https://juce.com/get-juce

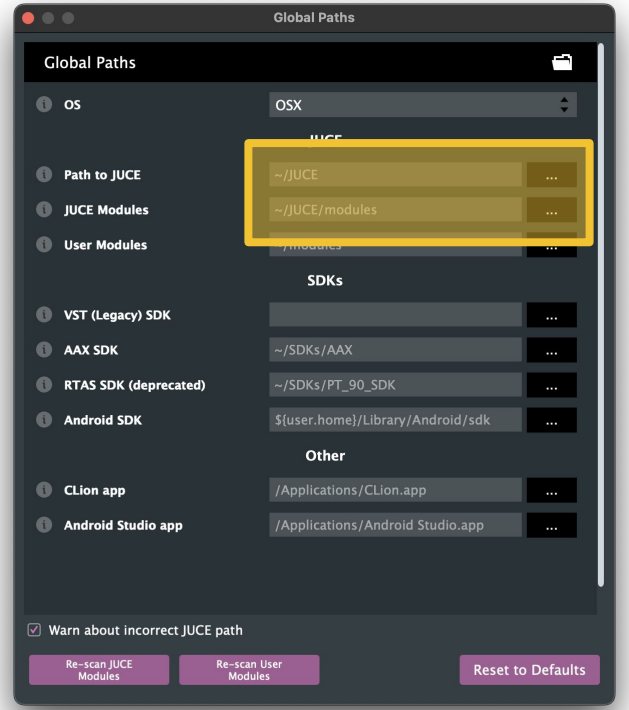
Installation: Download



1. Choose the OS, download, and unzip.
2. Move ‘JUCE’ folder to the path you want to keep it.
   - Usually ‘/Users/username/JUCE’ for macOS and ‘C:\JUCE’ for Windows.
3. Done.

Installation: Path Setting





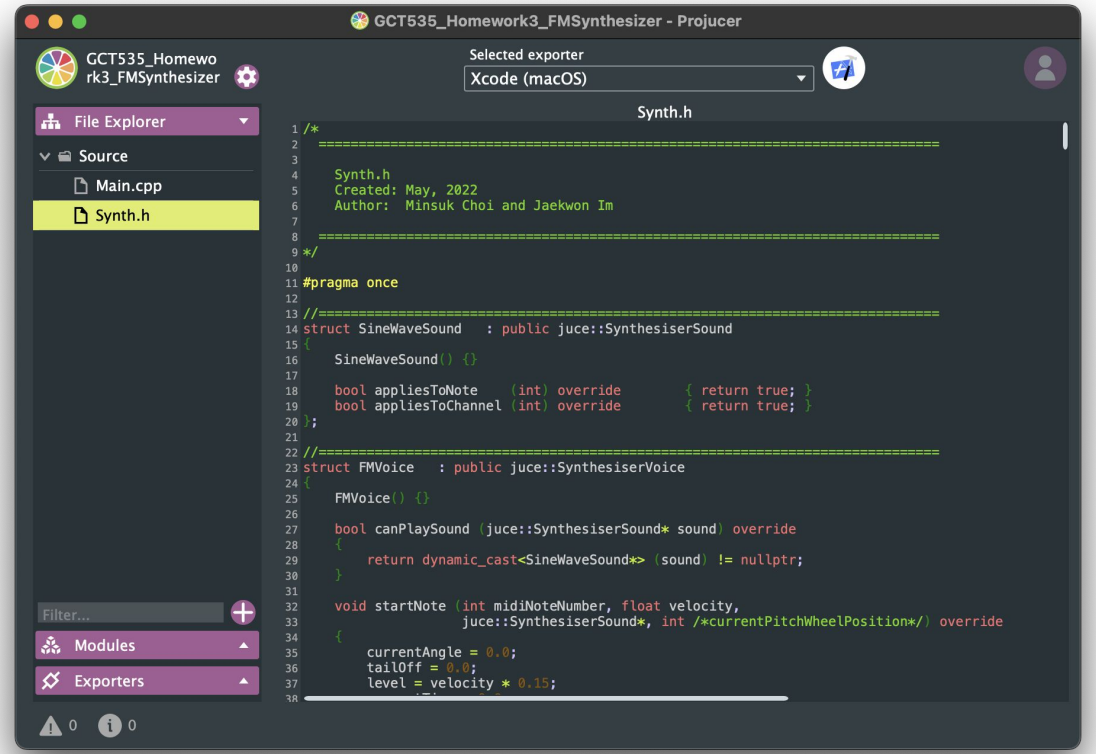
Please check the paths and set them to the path you installed JUCE

- Path to JUCE
- JUCE Modules

Setting Up JUCE: Exporters

- You'll need Xcode (macOS) or Visual Studio (Windows) as an exporter to build the source code.
- Links are here.
  - Visual Studio: https://visualstudio.microsoft.com/ko/downloads/
  - Xcode: https://developer.apple.com/download/
- Homework 3 is tested on Xcode Version 13.3.1 and Visual Studio 2019 16.11.13.

Setting Up JUCE



Select the exporter you want to use.

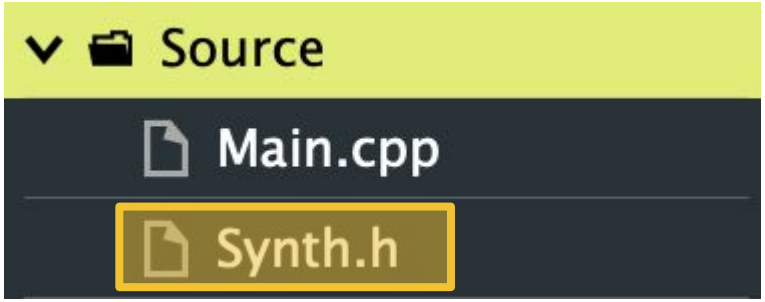
Run it to build the code.

Components: Synth.h and Main.cpp

.h files are header files for declaration of elements for program such as variable, classes, etc..

.cpp files are source files that actually do things with the elements that we've declared in header files.

Components: Synth.h and Main.cpp



Synth.h is the file you have to work with.

Realtime Audio Processing

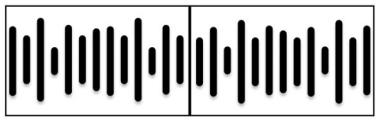
Non-Realtime Audio Processing
(So far in this course…)

Make Time samples array

Calculate at once

Realtime Audio Processing
(In this homework…)

Calculate each Time sample

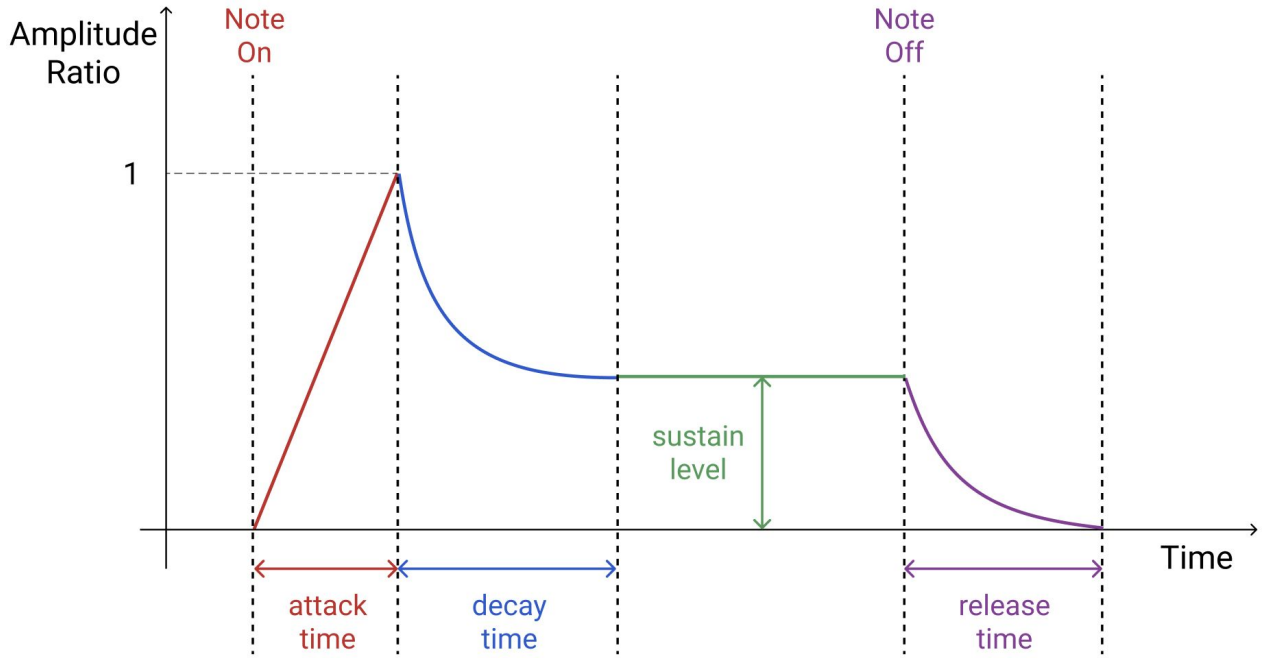


Realtime Audio Processing

```
float getCurrentSample (    float carrierAmplitude,
                            float carrierAttackTime, float carrierDecayTime,
                            float carrierSustainLevel, float carrierReleaseTime,
                            float modulatorAmplitude, float modulatorFreqRatio,
                            float modulatorAttackTime, float modulatorDecayTime,
                            float modulatorSustainLevel, float modulatorReleaseTime,
                            bool isRelease
)
```

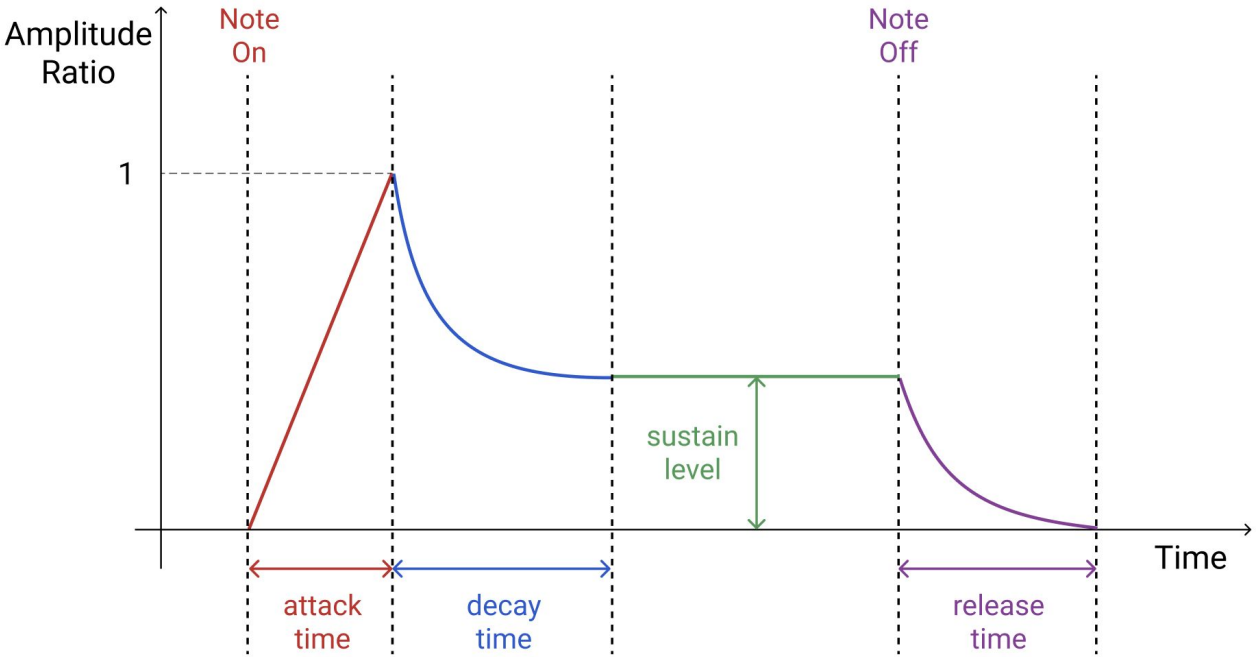
```
void renderNextBlock (  juce::AudioSampleBuffer& outputBuffer, int startSample, int numSamples,
                        float carrierAmplitude,
                        float carrierAttackTime, float carrierDecayTime,
                        float carrierSustainLevel, float carrierReleaseTime,
                        float modulatorAmplitude, float modulatorFreqRatio,
                        float modulatorAttackTime, float modulatorDecayTime,
                        float modulatorSustainLevel, float modulatorReleaseTime
)
```

ADSR: Overview



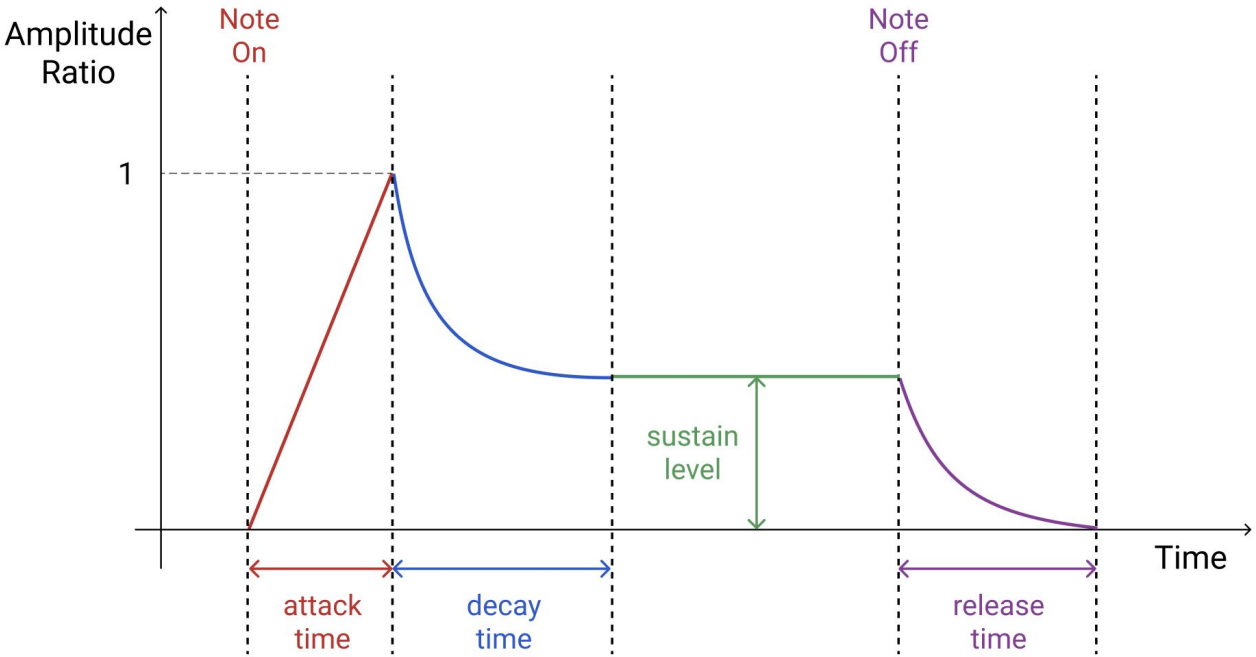
$$Sample = A_c * ADSR_c * \sin\left(2\pi f_c * t + A_m * ADSR_m * \sin(2\pi f_c * FreqRatio * t)\right)$$

ADSR: Attack



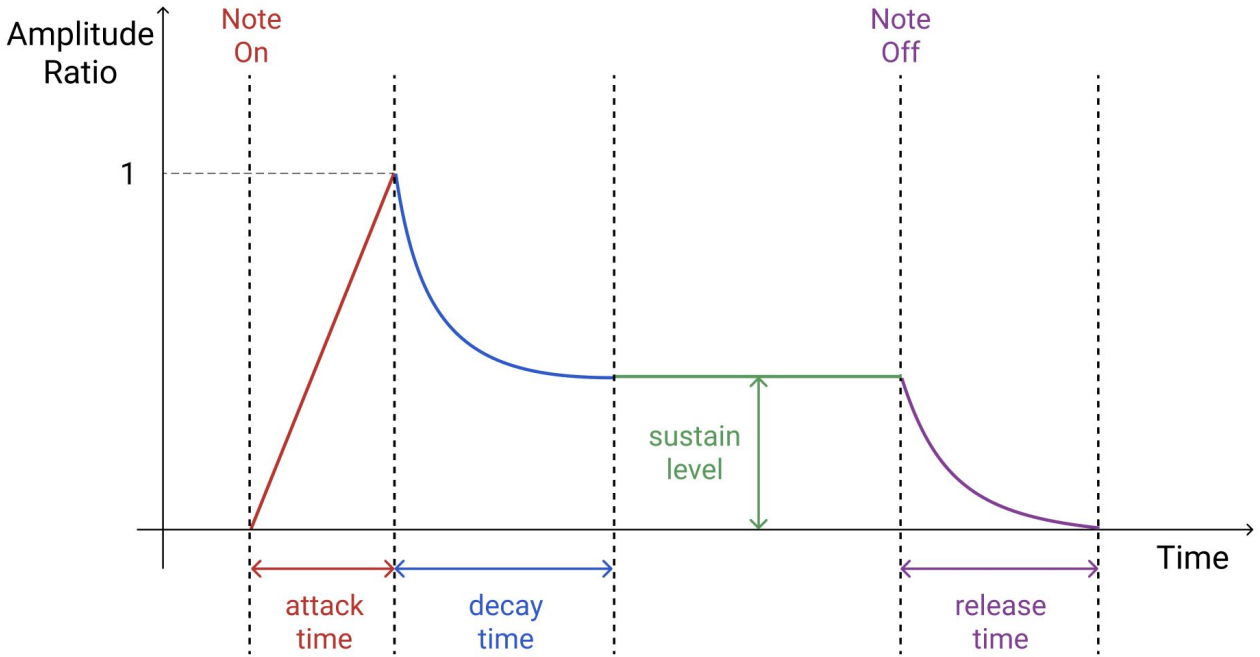
Attack time is the time it takes for a note to start and reach a maximum amplitude ratio which is 1 in our homework.

ADSR: Decay



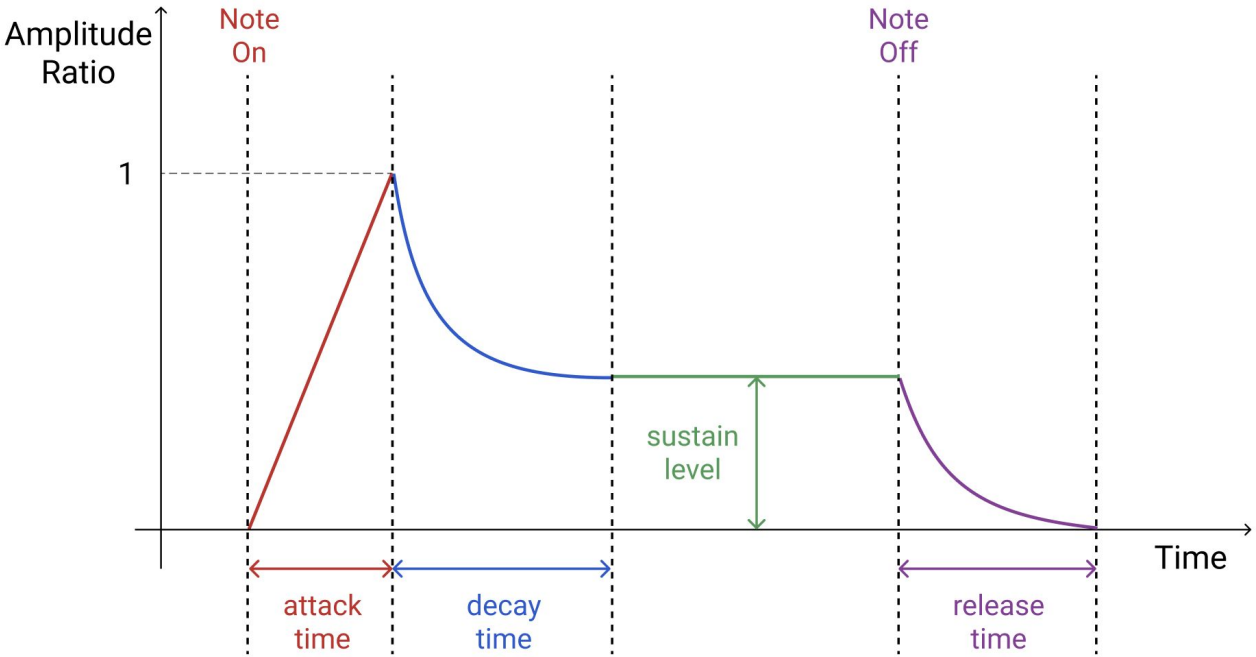
Decay time is the time it takes to reach the sustain level after the note reaches its peak level. In our homework, it should reduce exponentially.

ADSR: Sustain



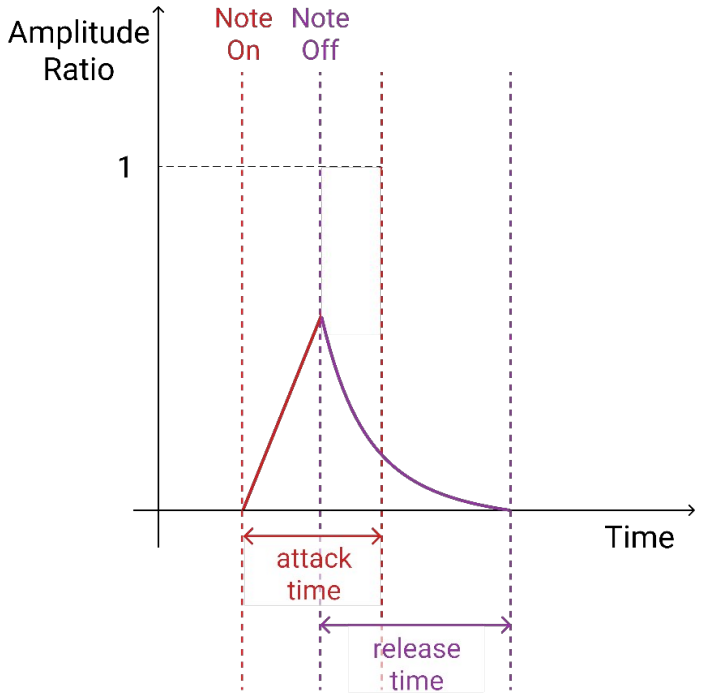
If note press time exceeds attack time + decay time, the amplitude of the note is fixed at the sustain level.

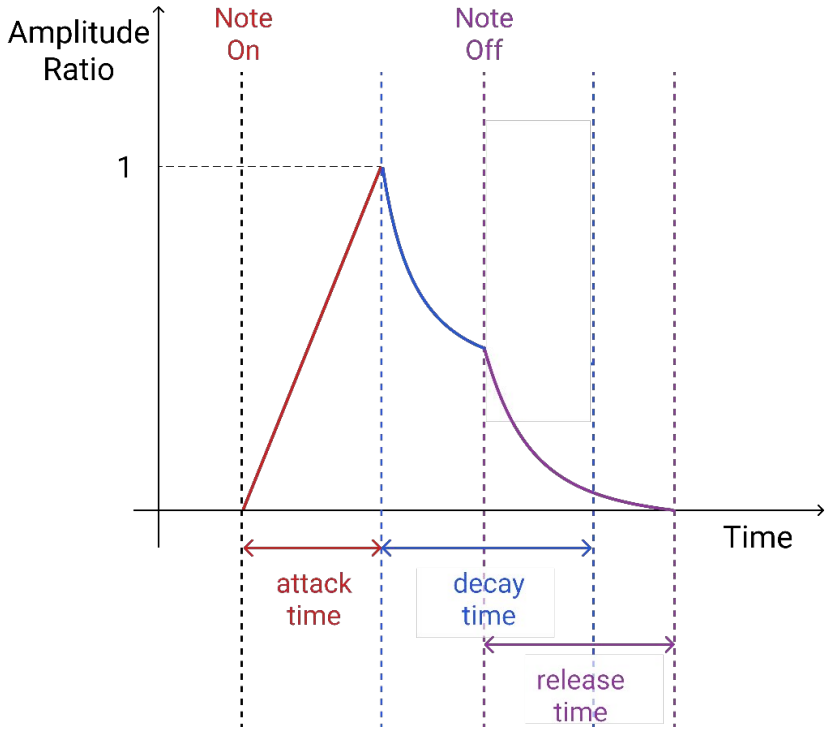
ADSR: Release



When the note is off, the amplitude decreases to zero during the release time. In our homework, it should reduce exponentially.

ADSR: Release Before Reaching Sustain





If the note is turned off during attack time or decay time, it decreases from the amplitude at that point to zero during the release time. In our homework, amplitude decreases exponentially at this time as well.

END